

(19) 日本国特許庁 (J P)

(12) 公開特許公報 (A)

(11) 特許出願公開番号  
特開2003-16410  
(P2003-16410A)

(43) 公開日 平成15年1月17日 (2003.1.17)

(51) Int.Cl. <sup>7</sup>	識別記号	F I	データシート <sup>*</sup> (参考)
G 0 6 K 19/07		B 4 2 D 15/10	5 2 1 2 C 0 0 j
B 4 2 D 15/10	5 2 1	C 0 6 K 17/00	D 5 B 0 3 j
G 0 6 F 9/44		19/00	N 5 B 0 5 8
9/54		G 0 6 F 9/06	6 4 0 B 5 B 0 7 6
G 0 6 K 17/00			6 2 0 K

審査請求 未請求 請求項の数13 O L (全 13 頁)

(21) 出願番号 特願2001-201807 (P2001-201807)

(22) 出願日 平成13年7月3日 (2001.7.3)

(71) 出願人 000002897

大日本印刷株式会社

東京都新宿区市谷加賀町一丁目1番1号

(72) 発明者 柴田 直人

東京都新宿区市谷加賀町一丁目1番1号

大日本印刷株式会社内

(72) 発明者 入澤 和義

東京都新宿区市谷加賀町一丁目1番1号

大日本印刷株式会社内

(74) 代理人 100091476

弁理士 志村 浩

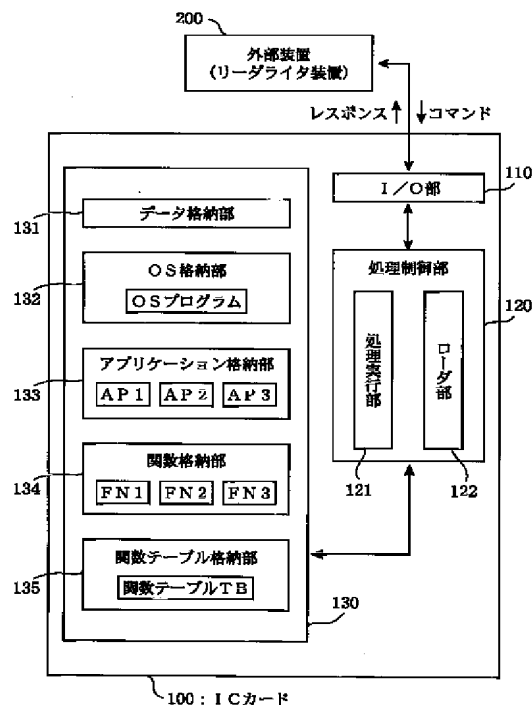
最終頁に続く

(54) 【発明の名称】 I Cカード

(57) 【要約】

【課題】 複数のプログラムから共通して利用可能な関数モジュールを、単体として容易に追加する。

【解決手段】 ロード部122に、アプリケーションおよび関数モジュールをロードする機能を設ける。新たにロードされたアプリケーションはアプリケーション格納部133に格納され、新たにロードされた関数モジュールは関数格納部134に格納される。ロードされた関数モジュールの関数名と実アドレスとの対応関係を示す関数テーブルTBが関数テーブル格納部135に作成される。アプリケーションからの関数呼び出しは、関数テーブルTBを参照して、実アドレスを認識することにより行われる。関数モジュールは、OSプログラムや他の関数モジュールからも呼び出される。



## 【特許請求の範囲】

【請求項1】 外部装置との間でコマンドおよびレスポンスをやりとりすることにより所定の処理を実行するICカードであって、  
 外部装置との通信を行うI/O部と、  
 データを格納するデータ格納部と、  
 OSプログラムを格納するOS格納部と、  
 アプリケーションプログラムを格納するアプリケーション格納部と、  
 関数モジュールを格納する関数格納部と、  
 前記関数格納部に格納されている関数モジュールの名称である関数名と、当該関数モジュールの実アドレスと、の対応関係を示す関数テーブルを格納する関数テーブル格納部と、  
 外部装置からロードコマンドとともにアプリケーションプログラムが与えられた場合に、与えられたアプリケーションプログラムを前記アプリケーション格納部に格納する処理と、外部装置からロードコマンドとともに関数モジュールが与えられた場合に、与えられた関数モジュールを前記関数格納部に格納するとともに当該格納した関数モジュールについての関数テーブルを前記関数テーブル格納部に格納する処理と、を前記OSプログラムに基づいて実行するロード部と、  
 外部装置から与えられたコマンドを、前記OSプログラムまたは前記アプリケーションプログラムに基づいて実行し、実行結果をレスポンスとして外部装置へと返す処理を行う処理実行部と、  
 を備え、  
 前記処理実行部は、  
 外部装置から与えられたアプリケーション選択コマンドに基づいて、もしくはOS格納部に格納されているOSプログラムで指示された初期設定に基づいて、前記アプリケーション格納部に格納されている複数のアプリケーションプログラムのうちの1つを選択する機能と、  
 外部装置からアプリケーション用コマンドが与えられた場合に、当該コマンドを現時点で選択されているアプリケーションプログラムに基づいて実行する機能と、  
 実行中のアプリケーションプログラムから、特定の関数名を指定した関数実行命令が与えられた場合に、前記関数テーブル格納部内の関数テーブルを参照することにより、当該特定の関数名に対応する実アドレスを認識し、前記実行中のアプリケーションプログラムを呼出元として、前記関数格納部内の前記実アドレスに格納されている関数モジュールを実行する機能と、  
 を有することを特徴とするICカード。

【請求項2】 請求項1に記載のICカードにおいて、  
 処理実行部が、実行中のOSプログラムから、特定の関数名を指定した関数実行命令が与えられた場合に、前記関数テーブル格納部内の関数テーブルを参照することにより、当該特定の関数名に対応する実アドレスを認識

し、前記実行中のOSプログラムを呼出元として、前記関数格納部内の前記実アドレスに格納されている関数モジュールを実行する機能を有することを特徴とするICカード。

【請求項3】 請求項1に記載のICカードにおいて、  
 処理実行部が、実行中の関数モジュールから、特定の関数名を指定した関数実行命令が与えられた場合に、前記関数テーブル格納部内の関数テーブルを参照することにより、当該特定の関数名に対応する実アドレスを認識し、前記実行中の関数モジュールを呼出元として、前記関数格納部内の前記実アドレスに格納されている関数を実行する機能を有することを特徴とするICカード。

【請求項4】 請求項1に記載のICカードにおいて、  
 外部装置から「カプセル化された関数実行命令を含むOS用もしくはアプリケーション用コマンド」が与えられた場合に、処理実行部が、当該コマンドをOSプログラムもしくは現時点で選択されているアプリケーションプログラムに基づいて実行し、関数テーブル格納部内の関数テーブルを参照することにより、前記関数実行命令によって示されている特定の関数名に対応する実アドレスを認識し、前記OSプログラムもしくは現時点で選択されているアプリケーションプログラムを呼出元として、前記関数格納部内の前記実アドレスに格納されている関数モジュールを実行する機能を有することを特徴とするICカード。

【請求項5】 請求項1～4のいずれかに記載のICカードにおいて、  
 処理実行部が、特定の関数モジュールを実行する際に、関数の呼出元となるプログラムにおけるプログラムカウンタを退避する処理を行い、前記特定の関数モジュールの実行後に、退避したプログラムカウンタに応じた位置から、前記呼出元となるプログラムの実行に復帰することを特徴とするICカード。

【請求項6】 請求項1～5のいずれかに記載のICカードにおいて、  
 外部装置からロードコマンドおよび関数モジュールとともに、当該関数モジュールの呼出元を限定する呼出元限定情報が与えられた場合に、ロード部が、当該関数モジュールについての前記呼出元限定情報を関数テーブルの一部として格納する処理を行う機能を有し、  
 処理実行部が、前記関数テーブル内の呼出元限定情報によって示された限定条件の下で関数モジュールの実行を行うことを特徴とするICカード。

【請求項7】 請求項1～5のいずれかに記載のICカードにおいて、  
 ロード部が、現時点でICカード内に格納されている呼出元となるプログラムから呼び出しが可能な関数名を外部装置にレスポンスとして報知する機能を有することを特徴とするICカード。

【請求項8】 請求項1～5のいずれかに記載のICカ

ードにおいて、

外部装置からロードコマンドとともに関数モジュールが与えられた場合に、ロード部が、与えられた関数モジュールを呼び出すための呼出元となるプログラムが現時点でICカード内に格納されているか否かを調査し、格納されていない場合には、与えられたロードコマンドに対してエラーを示すレスポンスを外部装置に返す機能を有することを特徴とするICカード。

【請求項9】 請求項1～5のいずれかに記載のICカードにおいて、

外部装置からロードコマンドとともに関数モジュールが与えられた場合に、ロード部が、所定のセキュリティ条件が満足されているか否かを調査し、満足されていない場合には、与えられたロードコマンドに対してエラーを示すレスポンスを外部装置に返す機能を有することを特徴とするICカード。

【請求項10】 請求項1～5のいずれかに記載のICカードにおいて、

外部装置からロードコマンドとともに暗号化された関数モジュールが与えられた場合に、ロード部が、前記暗号化された関数モジュールを復号化してから関数格納部に格納する機能を有することを特徴とするICカード。

【請求項11】 請求項1～5のいずれかに記載のICカードにおいて、

外部装置からロードコマンドとともに署名付きの関数モジュールが与えられた場合に、ロード部が、前記署名が正しいことを検証した後に、前記関数モジュールを関数格納部に格納する機能を有することを特徴とするICカード。

【請求項12】 請求項1～5のいずれかに記載のICカードにおいて、

ロード部が、外部装置からのコマンドまたはアプリケーションプログラムもしくはOSプログラムによる命令に基づいて、関数格納部内に格納されている関数モジュールの利用を一時停止させる機能と、この一時停止からの復帰をさせる機能と、を有することを特徴とするICカード。

【請求項13】 請求項1～5のいずれかに記載のICカードにおいて、

ロード部が、外部装置からのコマンドまたはアプリケーションプログラムもしくはOSプログラムによる命令に基づいて、関数格納部内に格納されている関数モジュールを削除、置換、または変更する機能を有することを特徴とするICカード。

【発明の詳細な説明】

【0001】

【発明の属する技術分野】本発明はICカードに関し、特に、外部装置との間でコマンドおよびレスポンスをやりとりすることにより所定の処理を実行するICカードに関する。

【0002】

【従来の技術】携帯可能な情報記録媒体として、ICカードが様々な分野で利用されるようになってきている。現在、一般に普及しているICカードは、CPUを内蔵し、外部装置との間でコマンドおよびレスポンスをやりとりすることにより所定の処理を実行する機能を有している。どのようなコマンドを与えると、ICカード内部でどのような処理が実行されるかは、ICカードに組み込まれたOSプログラムおよびアプリケーションプログラムによって定まるため、通常、個々の用途に応じた処理機能を有する専用のアプリケーションプログラムがICカード内に組み込まれる。最近では、半導体回路の集積度がかなり高まり、ICカードに内蔵されるメモリ容量も飛躍的に向上してきたため、複数のアプリケーションプログラムを予め組み込んだ多用途向けのICカードも一般化してきている。また、発行後に、必要に応じて新たなアプリケーションプログラムを追加する機能を備えたICカードも広く利用されている。たとえば、特開平10-79000号公報には、バージョンアップなどのためにプログラムを追加する機能をもったICカードが開示されている。

【0003】

【発明が解決しようとする課題】上述したように、新たなアプリケーションプログラムを、必要に応じて、その都度、追加することができるICカードは、既に利用されている。しかしながら、このようなアプリケーションプログラムは、あくまでも単体で機能するプログラムであり、従来のICカードにおいて追加できるプログラムは、このような単体で機能するプログラムに限られている。ところが、実際には、関数モジュールのように、単体では機能せずに、OSプログラムやアプリケーションプログラムなどから1つのルーチンとして利用されるプログラムも少なくない。従来、このような関数モジュールは、OSプログラムやアプリケーションプログラムの中に組み込まれており、関数モジュールを単体としてICカードに追加することは困難である。もちろん、アプリケーションプログラム自体に、このような関数モジュールを追加する何らかの機能をもたせておけば、当該アプリケーションからは、追加した関数モジュールを利用することはできる。しかしながら、個々のアプリケーションごとにこのような機能を用意することは非効率的であり、あまり実用的な方法とは言えない。

【0004】そこで本発明は、複数のプログラムから共通して利用可能な関数モジュールを、単体として容易に追加することが可能なICカードを提供することを目的とする。

【0005】

【課題を解決するための手段】(1) 本発明の第1の態様は、外部装置との間でコマンドおよびレスポンスをやりとりすることにより所定の処理を実行するICカード

において、外部装置との交信を行うI/O部と、データを格納するデータ格納部と、OSプログラムを格納するOS格納部と、アプリケーションプログラムを格納するアプリケーション格納部と、関数モジュールを格納する関数格納部と、関数格納部に格納されている関数モジュールの名称である関数名と、当該関数モジュールの実アドレスと、の対応関係を示す関数テーブルを格納する関数テーブル格納部と、外部装置からロードコマンドとともにアプリケーションプログラムが与えられた場合に、与えられたアプリケーションプログラムをアプリケーション格納部に格納する処理と、外部装置からロードコマンドとともに関数モジュールが与えられた場合に、与えられた関数モジュールを関数格納部に格納するとともに当該格納した関数モジュールについての関数テーブルを関数テーブル格納部に格納する処理と、をOSプログラムに基づいて実行するロード部と、外部装置から与えられたコマンドを、OSプログラムまたはアプリケーションプログラムに基づいて実行し、実行結果をレスポンスとして外部装置へと返す処理を行う処理実行部と、を設け、処理実行部が、外部装置から与えられたアプリケーション選択コマンドに基づいて、もしくはOS格納部に格納されているOSプログラムで指示された初期設定に基づいて、アプリケーション格納部に格納されている複数のアプリケーションプログラムのうちの1つを選択する機能と、外部装置からアプリケーション用コマンドが与えられた場合に、当該コマンドを現時点で選択されているアプリケーションプログラムに基づいて実行する機能と、実行中のアプリケーションプログラムから、特定の関数名を指定した関数実行命令が与えられた場合に、関数テーブル格納部内の関数テーブルを参照することにより、当該特定の関数名に対応する実アドレスを認識し、実行中のアプリケーションプログラムを呼出元として、関数格納部内の実アドレスに格納されている関数モジュールを実行する機能と、を行うことができるようにしたものである。

【0006】(2) 本発明の第2の態様は、上述の第1の態様に係るICカードにおいて、処理実行部が、実行中のOSプログラムから、特定の関数名を指定した関数実行命令が与えられた場合に、関数テーブル格納部内の関数テーブルを参照することにより、当該特定の関数名に対応する実アドレスを認識し、実行中のOSプログラムを呼出元として、関数格納部内の実アドレスに格納されている関数モジュールを実行するようにしたものである。

【0007】(3) 本発明の第3の態様は、上述の第1の態様に係るICカードにおいて、処理実行部が、実行中の関数モジュールから、特定の関数名を指定した関数実行命令が与えられた場合に、関数テーブル格納部内の関数テーブルを参照することにより、当該特定の関数名に対応する実アドレスを認識し、実行中の関数モジュール

を呼出元として、関数格納部内の実アドレスに格納されている関数を実行するようにしたものである。

【0008】(4) 本発明の第4の態様は、上述の第1の態様に係るICカードにおいて、外部装置から「カプセル化された関数実行命令を含むOS用もしくはアプリケーション用コマンド」が与えられた場合に、処理実行部が、当該コマンドをOSプログラムもしくは現時点で選択されているアプリケーションプログラムに基づいて実行し、関数テーブル格納部内の関数テーブルを参照することにより、関数実行命令によって示されている特定の関数名に対応する実アドレスを認識し、OSプログラムもしくは現時点で選択されているアプリケーションプログラムを呼出元として、関数格納部内の実アドレスに格納されている関数モジュールを実行するようにしたものである。

【0009】(5) 本発明の第5の態様は、上述の第1～第4の態様に係るICカードにおいて、処理実行部が、特定の関数モジュールを実行する際に、関数の呼出元となるプログラムにおけるプログラムカウンタを退避する処理を行い、特定の関数モジュールの実行後に、退避したプログラムカウンタに応じた位置から、呼出元となるプログラムの実行に復帰できるようにしたものである。

【0010】(6) 本発明の第6の態様は、上述の第1～第5の態様に係るICカードにおいて、外部装置からロードコマンドおよび関数モジュールとともに、当該関数モジュールの呼出元を限定する呼出元限定情報が与えられた場合に、ロード部が、当該関数モジュールについての呼出元限定情報を関数テーブルの一部として格納する処理を行い、処理実行部が、関数テーブル内の呼出元限定情報によって示された限定条件の下で関数モジュールの実行を行うようにしたものである。

【0011】(7) 本発明の第7の態様は、上述の第1～第5の態様に係るICカードにおいて、ロード部が、現時点でICカード内に格納されている呼出元となるプログラムから呼び出しが可能な関数名を外部装置にレスポンスとして報知するようにしたものである。

【0012】(8) 本発明の第8の態様は、上述の第1～第5の態様に係るICカードにおいて、外部装置からロードコマンドとともに関数モジュールが与えられた場合に、ロード部が、与えられた関数モジュールを呼び出すための呼出元となるプログラムが現時点でICカード内に格納されているか否かを調査し、格納されていない場合には、与えられたロードコマンドに対してエラーを示すレスポンスを外部装置に返すようにしたものである。

【0013】(9) 本発明の第9の態様は、上述の第1～第5の態様に係るICカードにおいて、外部装置からロードコマンドとともに関数モジュールが与えられた場合に、ロード部が、所定のセキュリティ条件が満足され

ているか否かを調査し、満足されていない場合には、与えられたロードコマンドに対してエラーを示すレスポンスを外部装置に返すようにしたものである。

【0014】(10) 本発明の第10の態様は、上述の第1～第5の態様に係るICカードにおいて、外部装置からロードコマンドとともに暗号化された関数モジュールが与えられた場合に、ロード部が、暗号化された関数モジュールを復号化してから関数格納部に格納するようにしたものである。

【0015】(11) 本発明の第11の態様は、上述の第1～第5の態様に係るICカードにおいて、外部装置からロードコマンドとともに署名付きの関数モジュールが与えられた場合に、ロード部が、署名が正しいことを検証した後に、関数モジュールを関数格納部に格納するようにしたものである。

【0016】(12) 本発明の第12の態様は、上述の第1～第5の態様に係るICカードにおいて、ロード部が、外部装置からのコマンドまたはアプリケーションプログラムもしくはOSプログラムによる命令に基づいて、関数格納部内に格納されている関数モジュールの利用を一時停止させる機能と、この一時停止からの復帰をさせる機能と、を有するようにしたものである。

【0017】(13) 本発明の第13の態様は、上述の第1～第5の態様に係るICカードにおいて、ロード部が、外部装置からのコマンドまたはアプリケーションプログラムもしくはOSプログラムによる命令に基づいて、関数格納部内に格納されている関数モジュールを削除、置換、または変更する機能を有するようにしたものである。

【0018】

【発明の実施の形態】以下、本発明を図示する実施形態に基づいて説明する。

【0019】§1. 本発明に係るICカードの基本構成  
図1は、本発明の一実施形態に係るICカード100を、外部装置200に接続した状態を示すブロック図である。外部装置200は、一般に、リーダライタ装置と呼ばれている装置であり、通常は、パソコンなどに接続して利用される。ICカード100には、外部装置200と交信を行うためのI/O部110と、ICカード内で種々の処理や制御を行う処理制御部120と、種々のデータやプログラムを格納するメモリ部130とが内蔵されている。ICカード100と外部装置200との間の交信は、コマンドおよびレスポンスのやりとりによって行われる。すなわち、外部装置200側から所定のコマンドをICカード100に対して送信すると、このコマンドは、I/O部110において受信されて処理制御部120へと伝達される。処理制御部120は、このコマンドを解釈実行し、処理結果をレスポンスとしてI/O部110へと引き渡す。こうして、このレスポンスは、I/O部110から外部装置200へと送信される

ことになる。処理制御部120は、メモリ部130内に格納されているプログラムに基づいて、与えられたコマンドの実行を行い、このコマンドの実行に伴い、必要に応じて、メモリ部130内に格納されているデータへのアクセスを行うことになる。

【0020】ここに示す実施形態では、処理制御部120は、処理実行部121とロード部122とによって構成されている。また、メモリ部130は、データ格納部131、OS格納部132、アプリケーション格納部133、関数格納部134、関数テーブル格納部135の5つの格納部によって構成されている。もっとも、これらの各構成要素は、本発明を説明する便宜上、機能面に着目して捉えた構成要素であり、実際のハードウェア上の構成要素に対応しているものではない。実際には、処理制御部120は、ICカード内に埋め込まれたCPUやその他の論理装置によって実現され、メモリ部130は、ROM、RAM、EEPROMなどのメモリによって実現されることになる。また、処理制御部120を構成する処理実行部121およびロード部122は、いずれもCPU単独で実現できる構成要素ではなく、メモリ部130内に格納されているプログラムに基づく動作が前提となって実現される構成要素である。ただ、ここでは、本発明の構成を説明する上で、CPUによって実行される機能のうち、外部から与えられた一般的なコマンドを実行する構成要素を処理実行部121とし、外部から与えられた新たなアプリケーションプログラムや関数モジュールをメモリ部130内に格納するロードコマンドを実行する構成要素をロード部122として把握することにする。

【0021】メモリ部130の構成要素のうち、データ格納部131は、このICカード100内に格納すべき種々のデータを格納する部分であり、このICカード100の所有者であるユーザに関する個人データや取引データ、ICカード100の発行者に関するデータなどが格納される。これらのデータは、通常、不揮発性の書き込み可能なメモリであるEEPROMに格納される。また、ここでは、このデータ格納部131内に、いわゆるCPUの作業領域も設けられており、種々の変数、ポインタ、フラグなども、このデータ格納部131内に格納されるものとする。このような作業領域は、通常、揮発性の書き込み可能なメモリであるRAM内に設けられる。

【0022】一方、OS格納部132には、このICカード100の基本動作を記述したOSプログラムが格納される。現在、ICカード用のOSプログラムとしては、JavaCardやMULTOSなどが普及しており、処理制御部120は、このOSプログラムに基づいて基本的な動作を行うことになる。このようなOS格納部132は、ROM内のメモリ領域あるいはEEPROM内のメモリ領域に設けられる。これに対して、アプリ

ケーション格納部133には、アプリケーションプログラムが格納される。ここに示す実施形態の場合、アプリケーションプログラムはICカードの発行後に任意に追加することができる仕様となっており、アプリケーション格納部133はEEPROM内のメモリ領域に設けられている。図1に示す例では、3つのアプリケーションプログラムAP1、AP2、AP3が既にアプリケーション格納部133に格納された状態が示されている。

【0023】関数格納部134および関数テーブル格納部135は、本発明の特徴となるメモリ領域であり、EEPROM内のメモリ領域に設けられている。関数格納部134には、単独のプログラムとしては機能しない関数モジュールが格納される。図1に示す例では、3つの関数モジュールFN1、FN2、FN3が既に関数格納部134に格納された状態が示されている。関数テーブル格納部135には、この関数格納部134内に格納されている関数モジュールを利用するために必要となる関数テーブルTBが格納される。この関数テーブルTBには、関数格納部134に格納されている関数モジュールの名称である関数名と、当該関数モジュールの実アドレスと、の対応関係が示されている。

【0024】処理実行部121は、外部装置200から与えられた一般のコマンド（後述するように、「ロード部122によって実行されるべきロードコマンド」以外のコマンド）を、OS格納部132内のOSプログラムまたはアプリケーション格納部133内のアプリケーションプログラムに基づいて実行し、実行結果をレスポンスとして外部装置200へと返す処理を行うことになる。外部装置200側から与えるコマンドは、OSプログラムによって解釈実行可能なOS用コマンドと、特定のアプリケーションプログラムによって解釈実行可能なアプリケーション用コマンドと、に分類することができる。OS用コマンドの典型的な例は、アプリケーション選択コマンドである。アプリケーション選択コマンドは、アプリケーション格納部133内に格納されている複数のアプリケーションプログラムのうちの1つを選択するためのコマンドであり、外部装置200側から、特定のアプリケーションを指定したアプリケーション選択コマンドが与えられると、処理実行部121は、指定されたアプリケーションプログラムを選択状態にする。具体的には、当該アプリケーションプログラムが現在選択状態にあることを示す情報を、データ格納部131内に格納する。一方、外部装置200側から、アプリケーション用コマンドが与えられた場合、処理実行部121は、このコマンドをOSプログラムではなく、アプリケーションプログラムに基づいて処理することになる。このとき、処理実行部121は、いずれのアプリケーションプログラムが選択状態にあるかを判断した上で、現在選択状態にあるアプリケーションプログラムに基づいて、与えられたアプリケーション用コマンドを処理す

る。なお、ICカード100と外部装置200とを接続した初期状態においては、OSプログラムによって予め指定された特定のアプリケーションプログラムが、デフォルトの選択プログラムとなる初期設定がなされており、アプリケーション選択コマンドが与えられていない初期状態では、デフォルト指定された特定のアプリケーションプログラムが自動的に選択状態になっている。

【0025】結局、外部装置200側からコマンドが与えられた場合、当該コマンドがOS用コマンドであった場合には、OSプログラムに基づいて処理が実行され、当該コマンドがアプリケーション用コマンドであった場合には、現在選択状態にあるアプリケーションプログラムに基づいて処理が実行されることになる。いずれの場合にも、処理の結果を示すレスポンスが作成され、外部装置200側へ返される。実際には、処理制御部120は、JavaCardやMULTOSなどのOSプログラムのもとで動作するアプリケーションプログラムを解釈実行する仮想機械としての機能を果たすことになる。

【0026】このように、外部装置200側から与えられた一般のコマンドは、処理実行部121によって処理されるが、ロードコマンドについては、ロード部122による処理が行われる。このロードコマンドは、OSプログラムによって実行されるべきOS用コマンドの一種であり、新たなアプリケーションプログラムや新たな関数モジュールをメモリ部130内に格納するためのコマンドである。たとえば、図1に示す例では、アプリケーション格納部133内には、既に3つのアプリケーションプログラムAP1、AP2、AP3が格納された状態となっているが、ここに、更に新たなアプリケーションプログラムAP4を追加する必要がある場合には、外部装置200から所定のロードコマンドとともにアプリケーションプログラムAP4を構成するデータを、ICカード100側に与える操作を行えばよい。このようなロードコマンドが与えられると、ロード部122によって、このアプリケーションプログラムAP4をアプリケーション格納部133内に新たに格納する処理が行われ、これを実行するために必要な設定が行われる。このように、ICカードに、新たにアプリケーションプログラムを追加する処理は、従来の一般的なICカードにおいても既に実施されている公知の技術であり、ここでは、詳しい説明は省略する。

【0027】本発明の特徴は、ロード部122に、アプリケーションプログラムの追加処理機能だけではなく、単体としては機能しない関数モジュールの追加処理機能を設けた点にある。たとえば、図1に示す例では、関数格納部134内には、既に3つの関数モジュールFN1、FN2、FN3が格納された状態となっているが、ここに、更に新たな関数モジュールFN4を追加する必要がある場合には、外部装置200から所定のロードコマンドとともに関数モジュールFN4を構成するデー

タをICカード100側に与える操作を行えばよい。このようなロードコマンドが与えられると、ロード部122によって、この関数モジュールFN4を関数格納部134内に新たに格納する処理が行われ、更に、この新たに格納した関数モジュールFN4についての情報を関数テーブル格納部135内の関数テーブルTBにつけ加える処理が行われることになる。もちろん、このようなアプリケーションプログラムの追加格納処理や、関数モジュールの追加格納処理は、OSプログラムに基づいて実行されるべき処理であり、ロード部122にこのような処理を実行させるためには、OS格納部132内に用意されたOSプログラムに、そのような処理を実行させるための手順を記述しておく必要がある。

【0028】§2. 本発明に係るICカードにおける関数の実行

さて、これまで本発明に係るICカードの基本構成について述べた。ここでは、このICカードにおける関数の実行形態について説明を行う。一般に、JavaCardやMULTOSなど、一般的なICカード用OSプログラムのもとで動作するアプリケーションプログラムは、CPUに依存しない言語で記述されているため、アプリケーションプログラム自体では、プログラムやファイルを選択するコマンドを取り扱うことが困難である。本発明の狙いは、このように、CPUに依存しない言語で記述されているアプリケーションプログラムを実行するICカードにおいて、単体としては機能しない関数モジュールを容易に追加することができるようにする点にある。そのためには、関数モジュールを追加する処理および関数モジュールを呼び出す処理を行う機能を、OSプログラムによって提供する必要がある。関数モジュールを追加する処理が、OSプログラムの下で、ロード部122によって実行されることは、既に§1で説明したとおりである。ここでは、関数モジュールを呼び出す処理（関数を実行するための処理）が、OSプログラムの下で、処理実行部121によってどのように実行されるかについて説明する。

【0029】まず、従来の一般的なICカードにおける関数モジュールの取り扱いと、本発明に係るICカードにおける関数モジュールの取り扱いとの概念的な相違を、図2および図3を参照して説明しよう。図2は、従来のICカードにおける関数モジュールの取り扱いを示すブロック図である。図示の例では、2つのアプリケーションプログラムAP1、AP2がブロックとして示されている。また、各アプリケーションプログラムの中には、それぞれ関数モジュールFN1、FN2が組み込まれている。このように、従来のICカードの場合、アプリケーションプログラムが関数を利用する場合、当該関数モジュールは、アプリケーションプログラムの中に記述され、アプリケーションプログラムと一体化された形式になる。したがって、ロード部122によって、アプ

リケーションプログラムをアプリケーション格納部133内に格納した場合、当該アプリケーションプログラムで利用する関数モジュールも、このアプリケーションプログラムと一体となってアプリケーション格納部133内に格納されることになる。このように、従来のICカードでは、関数モジュールがアプリケーションプログラムの一部として融合した形になっていたため、OSプログラムには、関数モジュールを単独で取り扱う仕組みが用意されていない。

【0030】これに対して、本発明の基本思想では、図3に示す例のように、個々の関数モジュールをアプリケーションプログラム本体とは別個のプログラムとして、単独で取り扱う仕組みを用意しておき、必要に応じて必要な関数モジュールを、その都度、呼び出して利用するという手法を採る。図3の例の場合、2つのアプリケーションプログラムAP1、AP2と、3つの関数モジュールFN1、FN2、FN3とが用意されており、アプリケーションプログラムAP1から関数モジュールFN1が呼び出され、アプリケーションプログラムAP2から関数モジュールFN2が呼び出されている様子が示されている。関数モジュールFN1の本体は、アプリケーションプログラムAP1内には含まれておらず、別個独立したプログラムとして関数格納部134内に格納されており、関数モジュールFN2の本体は、アプリケーションプログラムAP2内には含まれておらず、こちらも別個独立したプログラムとして関数格納部134内に格納されている。

【0031】このように、本発明の場合、アプリケーションプログラムも関数モジュールも、ロード部122によって独立してロードされるプログラムであるという点では同じである。ただ、アプリケーションプログラムは、単体で何らかの機能を果たすプログラムであるのに対し、関数モジュールは、単体では機能せず、他のプログラムから呼び出されることによってその役目を果たすプログラムである、という点で相違しており、両者のこの相違点は、外部装置200側からの選択コマンドによる選択対象になるか否かという点における相違に結びつくことになる。すなわち、アプリケーションプログラムは、前述したように、外部装置200側からアプリケーション選択コマンドを与えることにより選択することができ、外部から明示的に特定のアプリケーションの実行を指示することができるが、関数モジュールは、このような外部からの選択コマンドにより選択して実行させることはできず、あくまでも他のプログラムから呼び出して利用することが前提となる。

【0032】アプリケーションプログラムから、特定の関数モジュールを呼び出すには、呼出元となるアプリケーションプログラム内に、特定の関数の実行を指示する関数実行命令を記述しておくようにすればよい。たとえば、図4に示すように、アプリケーションプログラムA

P1と、関数モジュールFN1、FN2とがロードされている場合を考えてみよう。ここでは、アプリケーションプログラムAP1は、アプリケーション格納部133内のアドレスAdd1～Add5の領域に格納されており、関数モジュールFN1は、関数格納部134内のアドレスAdd6～Add7の領域に格納されており、関数モジュールFN2は、関数格納部134内のアドレスAdd8～Add9の領域に格納されているものとする。図示の例では、アプリケーションプログラムAP1には、3か所に関数実行命令（この例では、「CALL FN1」のような文字列）が含まれている。これらの関数実行命令は、特定の関数名を指定して、これを実行する旨の命令であり、たとえば、アドレスAdd2に記述された「CALL FN1」なる命令は、「FN1」なる関数名を指定した関数実行命令である。

【0033】既にS1で述べたように、ロード部122が新たな関数モジュールを関数格納部134内に格納する処理を行う際には、当該関数モジュールの名称である関数名と、当該関数モジュールの実アドレスと、の対応関係を示す関数テーブルTBを、関数テーブル格納部135に格納する処理も行われる。図4の右下に示す関数テーブルTBは、関数モジュールFN1およびFN2について作成された関数テーブルTBの具体例を示すものである。この例では、関数名FN1に対応する実アドレスはAdd6となっており、関数モジュールFN1が格納されている先頭実アドレスがAdd6であることが示されており、また、関数名FN2に対応する実アドレスはAdd8となっており、関数モジュールFN2が格納されている先頭実アドレスがAdd8であることが示されている。処理実行部121は、実行中のアプリケーションプログラムから、特定の関数名を指定した関数実行命令が与えられた場合に、関数テーブル格納部135内の関数テーブルTBを参照することにより、当該特定の関数名に対応する実アドレスを認識し、実行中のアプリケーションプログラムを呼出元として、関数格納部134内の認識した実アドレスに格納されている関数モジュールを実行することになる。

【0034】たとえば、アプリケーションプログラムAP1内のルーチンを実行中に、アドレスAdd2に記述された「CALL FN1」なる関数実行命令に遭遇した場合には、関数テーブルTBを参照することにより、関数名FN1に対応する実アドレスがアドレスAdd6であることを認識し、このアドレスAdd6に格納されている関数モジュールFN1のルーチンの実行（呼び出し）を行うことになる。一般に、関数モジュールは、呼出元から与えられた何らかの変数X（複数の場合もあるし、文字列の場合もある）に基づいて、関数値F(X)を演算してこれを呼出元へと返す機能を果たすプログラムである。したがって、実際には、このアプリケーションプログラムAP1のアドレスAdd2に至るまでのル

ーチンにおいて、変数Xとして何らかの値が用意され、アドレスAdd2以降のルーチンにおいて、関数モジュールFN1から返された関数値F(X)を利用した何らかの処理が行われることになる。ここで、アプリケーションプログラムから関数モジュールへの変数Xの引き渡しや、関数モジュールからアプリケーションプログラムへの関数値F(X)の引き渡しは、たとえば、レジスタなどを用いた公知の方法で行うことができるので、ここでは詳しい説明は省略する。

【0035】アプリケーションプログラムから関数モジュールを呼び出した後は、再びアプリケーションプログラムへと復帰することになるが、このような復帰は、プログラムカウンタの退避処理によって行うことができる。すなわち、処理実行部121は、関数モジュールFN1を実行する際に、この関数の呼出元となるアプリケーションプログラムAP1におけるプログラムカウンタ（アドレスAdd2を示すカウンタ）を退避する処理を行う。そして、呼出先となった関数モジュールFN1の実行が完了した後に、退避したプログラムカウンタに応じた位置（アドレスAdd2にある関数実行命令の次の命令の位置）から、呼出元となったアプリケーションプログラムAP1の実行に復帰することになる。

【0036】なお、図4に示すアプリケーションプログラムAP1のアドレスAdd4には、関数モジュールFN3についての関数実行命令が記述されているが、この図4に示す例では、関数モジュールFN3はまだ関数格納部134内には格納されていない。したがって、図示の例においてアドレスAdd4に記述された関数実行命令は、現時点では実行不能な命令であり、プログラムカウンタがアドレスAdd4を指し示した場合には、エラーが生じてしまう。実は、この図4に示すアプリケーションプログラムAP1におけるアドレスAdd4以降のルーチンは、将来の機能拡張用に予め付加されているルーチンであり、実際には、この時点では、このルーチンが実行されることはない。別言すれば、このルーチンが実行されるようなコマンドは、この時点では、ICカードに対して与えない、という前提で、外部装置200側からのアクセスが行われることになる。もちろん、将来、アプリケーションプログラムAP1の機能を拡張する必要が生じた場合には、ロードコマンドを用いて関数モジュールFN3を新たにICカード内に格納すれば、このアドレスAdd4以降のルーチンによる付加機能が利用できるようになる。

【0037】このように、本発明に係るICカードでは、必要に応じて、その都度、必要な関数モジュールを新規にロードして利用できるように、柔軟な利用形態が可能になる。また、同一の関数モジュールを複数のアプリケーションプログラムから共通して利用するような形態も可能になり、関数の効率的な利用を図ることもできるようになる。



【0038】§3. 本発明における関数モジュールの利用形態の変形例

以上、本発明に係るICカードを基本的な実施形態に基づいて説明したが、本発明は上述した基本的な実施形態に限定されるものではなく、この他にも種々の形態で実施可能である。ここでは、本発明における関数モジュールの利用形態の変形例のいくつかを項分けして紹介する。

【0039】(1) 上述の実施形態では、アプリケーションプログラムから関数モジュールを呼び出す例のみを示したが、関数モジュールの呼出元は、必ずしもアプリケーションプログラムである必要はなく、たとえば、OSプログラムを呼出元として特定の関数モジュールを呼び出すようなことも可能である。すなわち、処理実行部121は、実行中のOSプログラムから、特定の関数名を指定した関数実行命令が与えられた場合に、関数テーブル格納部135内の関数テーブルTBを参照することにより、当該特定の関数名に対応する実アドレスを認識し、実行中のOSプログラムを呼出元として、関数格納部134内の所定の実アドレスに格納されている関数モジュールを実行する処理を行えばよい。また、第1の関数モジュールを呼出元として更に第2の関数モジュールを呼び出すような、いわゆる入れ子式の関数呼び出しを行うことも可能である。すなわち、処理実行部121は、実行中の関数モジュールから、特定の関数名を指定した関数実行命令が与えられた場合に、関数テーブル格納部135内の関数テーブルTBを参照することにより、当該特定の関数名に対応する実アドレスを認識し、実行中の関数モジュールを呼出元として、関数格納部134内の所定の実アドレスに格納されている関数を実行する処理を行えばよい。

【0040】(2) このように、本発明では、関数格納部134内の関数モジュールは、ICカード内部のプログラム（アプリケーションプログラム、OSプログラム、他の関数モジュール）から呼び出されることが前提となっており、原則として、外部装置200側から関数格納部134内の関数モジュールを直接呼び出して利用することはできない。ただ、外部装置200側から与えるコマンドに工夫を施せば、あたかも外部装置200側から関数格納部134内の関数モジュールを間接的に呼び出して利用するような利用形態も可能である。具体的には、外部装置200側から、「カプセル化された関数実行命令を含むOS用もしくはアプリケーション用コマンド」をICカード100側に与えるようにすればよい。そして、処理実行部121は、このようなコマンドが与えられたときに、当該コマンドを、OSプログラムもしくは現時点で選択されているアプリケーションプログラムに基づいて実行し、関数テーブル格納部135内の関数テーブルTBを参照することにより、カプセル化されていた関数実行命令によって示されている特定の

関数名に対応する実アドレスを認識し、OSプログラムもしくは現時点で選択されているアプリケーションプログラムを呼出元として、関数格納部134内の所定の実アドレスに格納されている関数モジュールを実行する処理を行うようにすればよい。

【0041】図5は、このような処理を実行させるためのアプリケーション用コマンドの構成例を示すブロック図である。このコマンドは、命令コードを示す「EXECFN」なる文字列の部分と、当該命令コードに付随するデータとなる「CALLFN1, X=123」なる文字列の部分と、によって構成されている。ここで、後半の「CALL FN1, X=123」なる部分は、実際には、関数実行命令および変数値を示すコードであるが、カプセル化された状態となっており、全体として見れば、「EXECFN」なる命令コードのデータ部分という形式になっている。したがって、このようなアプリケーション用コマンドが外部装置200側から与えられると、一般のコマンドと同様に、OSプログラムもしくはその時点で選択状態となっているアプリケーションプログラムへと引き渡されることになる。そこで、OSプログラムもしくはアプリケーションプログラム側に、この「EXECFN」なるコマンドが与えられた場合に、そのデータ部に記述された「CALL FN1, X=123」なる文字列を関数実行命令および変数値を示すコードと解釈して、関数モジュールFN1の呼び出し処理を実行するルーチンを予め用意しておけば、関数モジュールFN1に変数値X=123を引き渡し、関数値F(123)を得ることができる。更に、このアプリケーションプログラム側に、得られた関数値F(123)をレスポンスとして返すルーチンを用意しておけば、図5に示すようなアプリケーション用コマンドに対するレスポンスとして、関数値F(123)が外部装置200側に返されることになる。

【0042】このような手法を採れば、実際には、ICカード内に格納されているプログラムから関数モジュールFN1を呼び出して利用していることになるが、実質的には、外部装置200側からのコマンドによって、関数モジュールFN1を利用するような取り扱いが可能になる。

【0043】(3) 図4に示す関数テーブルTBには、各関数モジュールについて、関数名と実アドレスとの対応関係しか示されていないが、更に、当該関数モジュールの呼出元を限定する呼出元限定情報を付加することも可能である。図6は、このような呼出元限定情報を付加した関数テーブルTBの一例を示す図である。たとえば、関数モジュールFN1については、呼出元限定情報として、AP1, AP2, OSなる情報が付加されているが、これは、関数モジュールFN1の呼出元を、アプリケーションプログラムAP1, AP2およびOSプログラムに限定することを示す情報であり、関数モジュール

ルFN2については、呼出元限定情報として、AP1、AP3なる情報が付加されているが、これは、関数モジュールFN2の呼出元を、アプリケーションプログラムAP1、AP3に限定することを示す情報である。このような呼出元限定情報は、個々の関数モジュールをロードする際に、外部装置200側から指定すればよい。ロード部122は、外部装置200から関数モジュールとともに、当該関数モジュールの呼出元を限定する呼出元限定情報が与えられた場合に、当該関数モジュールについての呼出元限定情報を関数テーブルの一部として格納する処理を行うことになる。

【0044】処理実行部121は、関数テーブルTBに呼出元限定情報が付加されていた場合には、この呼出元限定情報によって示された限定条件の下で、関数モジュールの実行を行うことになる。たとえば、図6に示す例の場合、アプリケーションプログラムAP2が関数モジュールFN2を呼び出すような処理を行ったとしても、呼出元限定情報による限定条件を満たしていないので、このような呼出処理は正常な処理とは認められず、エラー処理がなされることになる。このような呼出元限定情報を設定しておく、と、特定のアプリケーションプログラムと密接に結び付いた関数モジュールが、他のアプリケーションプログラムから利用されるのを防ぐことができる。

【0045】§4. 本発明における関数モジュールのロード時の変形例

本発明に係るICカードでは、ロード部122によって、アプリケーションプログラムのみならず、必要に応じて関数モジュールを新たにロードすることができることは既に述べたとおりである。ここでは、この関数モジュールのロードを行う際の変形例のいくつかを項分けして紹介する。

【0046】(1) 既に述べたように、本発明においてICカード内にロードされる関数モジュールは、ICカード内のプログラム（アプリケーションプログラム、OSプログラム、あるいは他の関数モジュール）から呼び出されて利用されることが前提となる。したがって、呼出元となるプログラムがICカード内に存在しないような関数モジュールは、ICカード内にロードしても利用価値はない。そこで、ロード部122に、現時点でICカード内に格納されている呼出元となるプログラムから呼び出しが可能な関数名を外部装置にレスポンスとして報知する機能をもたせておくのが好ましい。たとえば、図4に示すようなアプリケーションプログラムAP1のみがICカード内に格納されている状態では、このアプリケーションプログラムAP1内を調べることにより、3つの関数モジュールFN1、FN2、FN3のみが呼び出し可能であることが認識できる。そこで、ロード部122に、この3つの関数モジュールの関数名である「FN1、FN2、FN3」なる文字列を、外部装置200

側に報知する機能をもたせておけば、外部装置200側は、現時点でこのICカードにおいて利用価値のある関数モジュールは、この3つの関数モジュールである旨を認識することができ、誤って利用価値のない関数モジュールをロードするような作業が行われるのを防ぐことができる。ロード部122から外部装置200側への報知は、レスポンスという形式で行われるが、このレスポンスを得るためには、外部装置200側から、当該レスポンスを要求する旨のコマンドを与えるようにすればよい。あるいは、ICカード100と外部装置200とを接続した初期状態において、ICカードをリセットしたときにICカード側から返されるATR（Answer to Reset）の信号に、呼び出しが可能な関数名を示す情報をもたせておくようにしてもよい。

【0047】(2) 上述した工夫と類似した工夫として、外部装置200からロードコマンドとともに関数モジュールが与えられた場合に、ロード部122が、与えられた関数モジュールを呼び出すための呼出元となるプログラムが現時点でICカード内に格納されているか否かを調査し、格納されていない場合には、与えられたロードコマンドに対してエラーを示すレスポンスを外部装置200に返すような処理を行うようにしてもよい。この場合は、ICカード100側から積極的に、呼び出しが可能な関数名の報知が行われるわけではないが、もし誤って呼び出し不能な関数モジュールをロードするような作業が行われた場合には、エラーが発生することになる。たとえば、図4に示すようなアプリケーションプログラムAP1のみがICカード内に格納されている状態において、新たに関数モジュールFN4をロードするような作業が行われた場合（すなわち、外部装置200側から、関数モジュールFN4についてのロードコマンドが与えられた場合）は、アプリケーションプログラムAP1内を調べることにより、関数モジュールFN4の呼び出しが不能であることが認識できるので、このようなロードコマンドは実行されずに、エラーを示すレスポンスが返されることになる。

【0048】(3) これまでの実施形態では、新たに関数モジュールをロードする際のセキュリティ確保の問題については触れていなかったが、実用上は、外部装置200からロードコマンドとともに関数モジュールが与えられた場合に、ロード部122が、所定のセキュリティ条件が満足されているか否かを調査し、満足されていない場合には、与えられたロードコマンドに対してエラーを示すレスポンスを外部装置200に返すようにしておくのが好ましい。セキュリティ条件が満足されているか否かの判断は、既に公知の種々の方法のいずれを用いてもかまわない。

【0049】また、関数モジュールをロードする際には、関数モジュール本体に対する暗号化を行っておくようにするのが好ましい。この場合、外部装置200側の

暗号化アルゴリズムに応じた復号化アルゴリズムをＩＣカード１００側に用意しておき、外部装置２００からロードコマンドとともに暗号化された関数モジュールが与えられた場合に、ロード部１２２が、用意されている復号化アルゴリズムにより、暗号化された関数モジュールを復号化してから関数格納部１３４に格納する処理を行うようにすればよい。

【００５０】更に、関数モジュールをロードする際には、関数モジュール本体に署名を付加するようにし、ＩＣカード内に、この署名を検証するアルゴリズムを用意しておくのが好ましい。この場合、外部装置２００からロードコマンドとともに署名付きの関数モジュールが与えられた場合に、ロード部１２２が、検証用アルゴリズムを用いて、付加されてきた署名が正しいことを検証した後に、この関数モジュールを関数格納部１３４に格納する処理を行えばよい。

【００５１】(4) ロード部１２２には、更に付加的な機能を設けておくこともできる。たとえば、外部装置２００からのコマンドまたはアプリケーションプログラムもしくはＯＳプログラムによる命令に基づいて、関数格納部１３４内に格納されている特定の関数モジュールの利用を一時停止させる機能と、この一時停止からの復帰をさせる機能と、をロード部１２２に設けておくこともできる。この機能を利用すれば、一時的に特定の関数モジュールの利用を制限するような処理を容易に行うことができる。また、外部装置２００からのコマンドまたはアプリケーションプログラムもしくはＯＳプログラムによる命令に基づいて、関数格納部１３４内に格納されている関数モジュールを削除、置換、または変更する機能を設けておくこともできる。不要になった関数モジュールが発生した場合、これを削除すれば、その分のメモリ領域を開放することができる。また、新たな関数モジュールに置換したり、部分的に変更したりすることができれば、関数モジュールを最新のバージョンに保つことができるようになる。

【００５２】

【発明の効果】以上のとおり、本発明に係るＩＣカードによれば、複数のプログラムから共通して利用可能な関数モジュールを、単体として容易に追加することが可能になる。

【図面の簡単な説明】

【図１】本発明の一実施形態に係るＩＣカード１００を、外部装置２００に接続した状態を示すブロック図である。

【図２】従来のＩＣカードにおける関数モジュールの取り扱いを示すブロック図である。

【図３】本発明に係るＩＣカードにおける関数モジュールの取り扱いを示すブロック図である。

【図４】アプリケーションプログラムＡＰ１と、関数モジュールＦＮ１、ＦＮ２とがロードされている場合のＩＣカードのメモリ領域の構成を示すブロック図である。

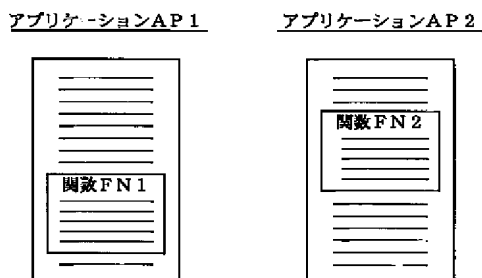
【図５】カプセル化された関数実行命令を含むアプリケーション用コマンドの構成例を示す図である。

【図６】呼出元限定情報を付加した関数テーブルＴＢの一例を示す図である。

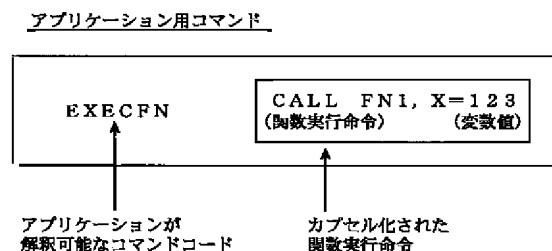
【符号の説明】

１００…ＩＣカード  
 １１０…Ｉ／Ｏ部  
 １２０…処理制御部  
 １２１…処理実行部  
 １２２…ロード部  
 １３０…メモリ部  
 １３１…データ格納部  
 １３２…ＯＳ格納部  
 １３３…アプリケーション格納部  
 １３４…関数格納部  
 １３５…関数テーブル格納部  
 Ａｄｄ１～Ａｄｄ９…実アドレス  
 ＡＰ１～ＡＰ３…アプリケーションプログラム  
 ＦＮ１～ＦＮ３…関数モジュール  
 ＴＢ…関数テーブル

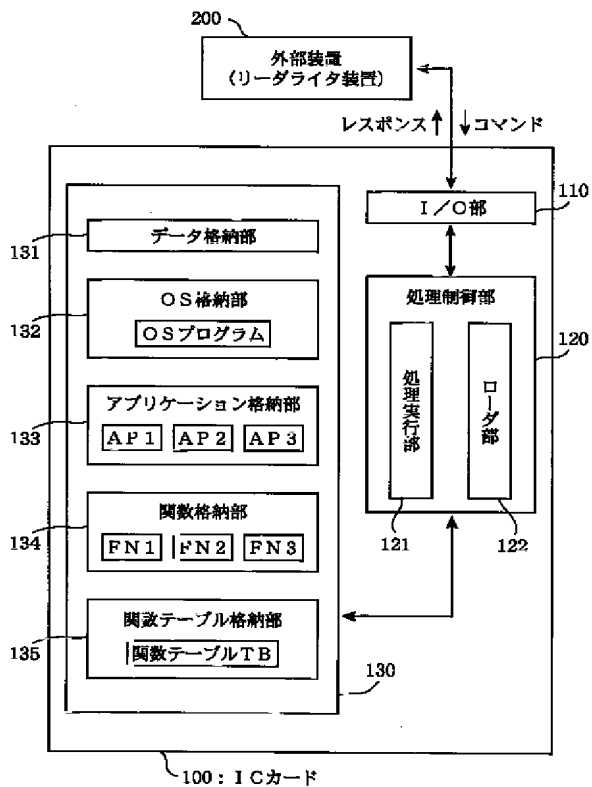
【図２】



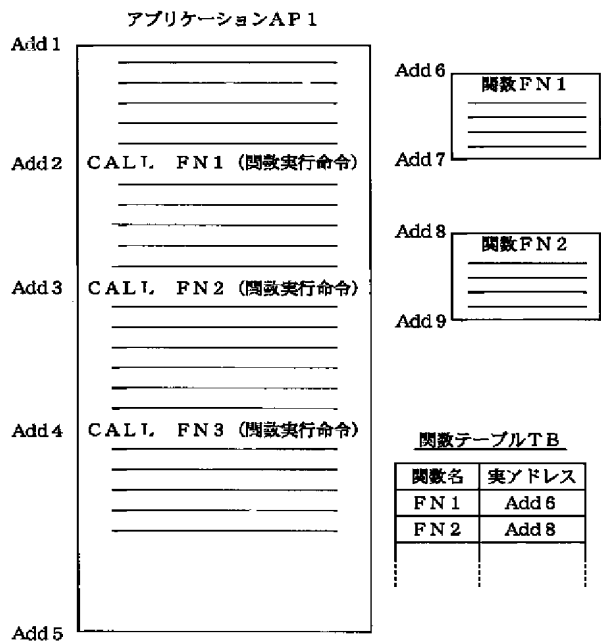
【図５】



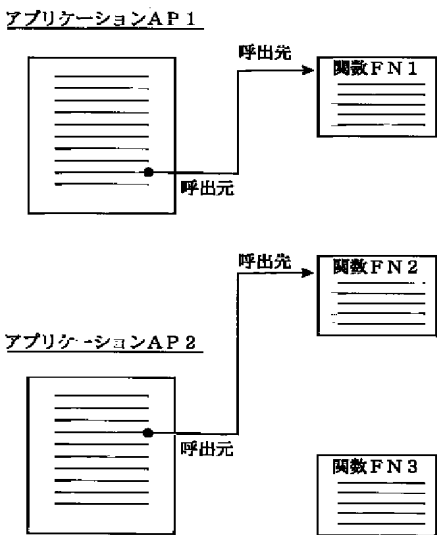
【図1】



【図4】



【図3】



【図6】

関数テーブルTB

関数名	実アドレス	呼出元限定情報
FN 1	Add 6	AP 1, AP 2, OS
FN 2	Add 8	AP 1, AP 3

フロントページの続き

(72)発明者 神力 哲夫  
東京都新宿区市谷加賀町一丁目1番1号  
大日本印刷株式会社内

Fターム(参考) 2C005 MA33 SA02 SA04 SA08 SA23  
5B035 AA06 BB09 CA11  
5B058 CA25 KA02 KA04 KA08 KA11  
YA20  
5B076 AB02 AB04 AB05 AB06